



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:2) mengemukakan bahwa, “Komputer merupakan peralatan elektronik yang bermanfaat, melaksanakan berbagai pekerjaan yang dilakukan oleh manusia ”.

Ikhsan dan Kurniawan (2015:13), “Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah.”

Berdasarkan pendapat di atas, dapat disimpulkan bahwa definisi dari komputer adalah sebuah alat elektronik yang dapat mengolah data sehingga dapat menghasilkan keluaran yang bermanfaat untuk memudahkan berbagai kegiatan manusia.

2.1.2 Pengertian Perangkat Lunak

Rosa dan Shalahuddin (2018: 2), perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).

Swara dan Febriadi (2016:8), “Perangkat lunak merupakan seluruh perintah yang digunakan untuk memproses informasi”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa perangkat lunak merupakan penghubung agar komputer melakukan pengolahan sesuai perintah dan kebutuhan dari *user*.



2.1.3 Pengertian Perangkat Keras

Mahyulida Munthe (2020:7) “Perangkat keras adalah komponen perangkat fisik pada komputer yang dapat dilihat dan disentuh secara langsung, yang setiap komponennya memiliki fungsi masing-masing untuk memaksimalkan kinerja komputer atau mengoperasikan sebuah komputer”.

2.1.4 Pengertian Data

Kristanto, Andri (2018:8) menyatakan bahwa, “Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi”.

“Data adalah kumpulan berupa pernyataan apa adanya dari fakta yang didapatkan dari hasil pengukuran atau pengamatan suatu variabel baik dalam bentuk angka, kata-kata, atau citra”. (<https://id.m.wikipedia.org/wiki/Data>)

Berdasarkan pendapat di atas, dapat disimpulkan bahwa data adalah kumpulan data baik berupa angka, kata-kata atau citra yang didapatkan melalui pengukuran atau pengamatan yang memerlukan pengolahan selanjutnya untuk menghasilkan informasi yang memberi manfaat bagi penerimanya.

2.1.5 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Sukamto dan Shalahuddin (2016:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2016:128-131) adalah sebagai berikut:

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).



2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.



2.2. Teori Khusus

2.2.1 *Unified Modelling Language (UML)*

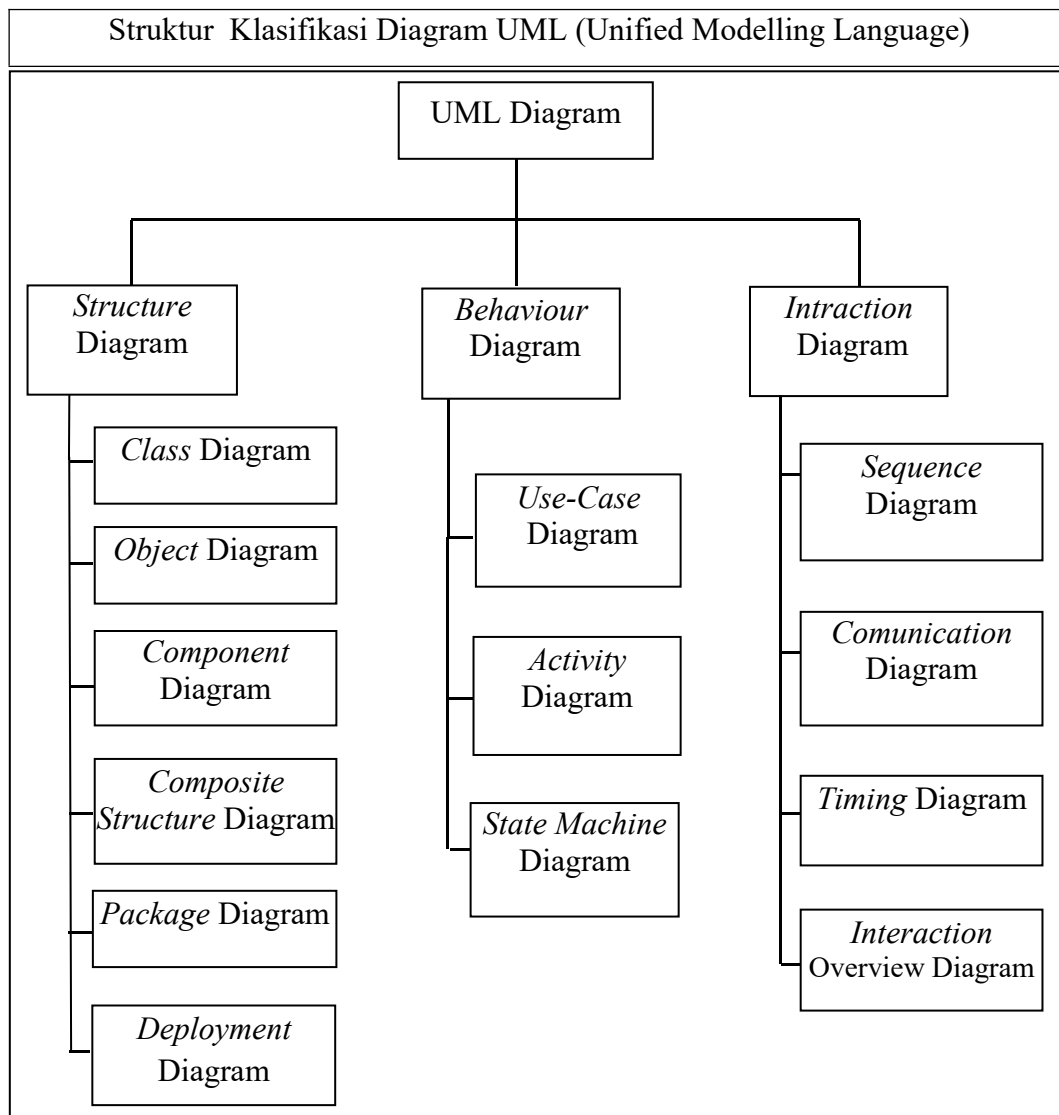
Widodo dalam Pranata dkk (2015:26), “*UML* adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak). *Artifact* dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya”.

Sukamto dan Shalahuddin (2016:133), “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

Dari pengertian diatas dapat disimpulkan bahwa *UML (Unified Modelling Language)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasikan hingga mendokumentasikan sebuah sistem pengembangan perangkat lunak berorientasi objek.

2.2.2 *Klasifikasi Diagram UML (Unified Modelling Language)*

Sukamto dan Shalahuddin (2018:140), “Pada *UML 2.3* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



Gambar 2.1 Klasifikasi Diagram UML (*Unified Modelling Language*)

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukanto dan Shalahuddin (2018:141) :

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

Interaction diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

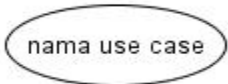
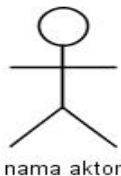




2.2.3. Use-Case Diagram

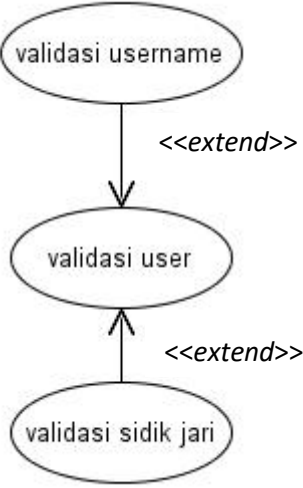
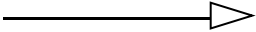
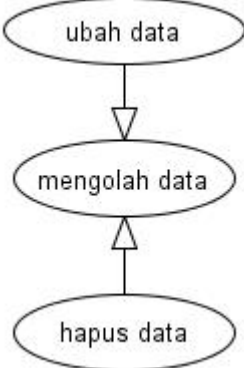
Widodo (dikuip Pranata dkk. 2015:26), *Use-case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use-case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Sukanto dan Shalahuddin (2016:155), “*Use-case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use-case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”. Adapun simbol-simbol yang digunakan dalam *use-case* adalah sebagai berikut:

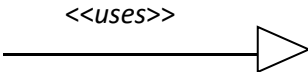
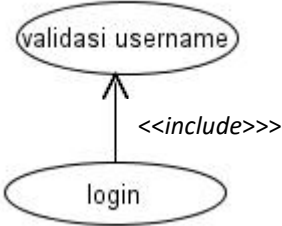
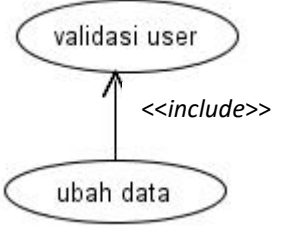
Tabel 2.1. Simbol-simbol *Use-Case* Diagram

No	Simbol	Deskripsi
1.		fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i> .
2.		orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.		komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
4.		relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya

Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

		 <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	Generalisasi <i>generalization</i> 	/ hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6.	menggunakan / include / uses> <<include>>	Relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :

Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

		<ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:  <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	---	---

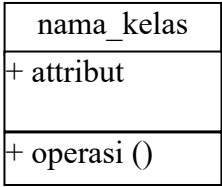



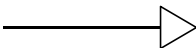
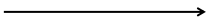
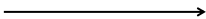
Sumber: Sukanto dan Shalahuddin (2016:156-158)

2.2.4. Class Diagram

Sukanto dan Shalahuddin (2016:141), Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
- Operasi atau model adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Tabel 2.2. Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan/ <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

(Sumber: Sukamto dan Shalahuddin (2016:144-147))




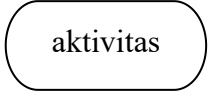
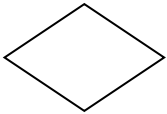

2.2.5. Activity Diagram

Sukanto dan Shalahuddin (2016:161), *Activity Diagram* atau Diagram Aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.


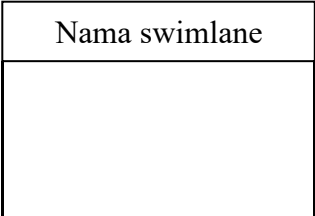
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak

Tabel 2.3. Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

**Lanjutan Tabel 2.3.** Simbol-simbol *Activity Diagram*

5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

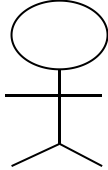
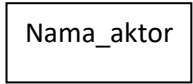

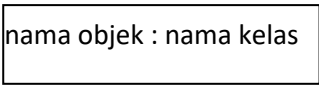

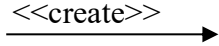
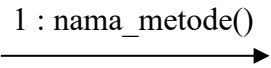
(Sumber: Sukamto dan Shalahuddin (2016:162-163))

2.2.6. *Sequence Diagram*

Sukamto dan Shalahuddin (2016:165), “Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat sekenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.”

Tabel 2.4. Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Aktor Atau</p>  <p>Nama_aktor tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2.	<p>Garis hidup/ <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	<p>pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.

Lanjutan Tabel 2.4. Simbol-simbol *Sequence Diagram*

7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destrory</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber: Sukamto dan Shalahuddin (2016:165-167))

2.3. Pengertian Judul

2.3.1 Pengertian Aplikasi

Juansyah, Andi (2015: 2), “Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta pengguna aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju”.

Koyuko dkk dalam Oki (2016), “Aplikasi adalah suatu *software* yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen, penambahan data dan beberapa tugas lainnya. *suite* aplikasi (*application suite*) adalah penggabungan dari berbagai macam aplikasi menjadi satu. salah satu contohnya *Microsoft Office* dan *Open Office* yaitu menggabungkan aplikasi *Microsoft word*, *Microsoft excel*, dan beberapa aplikasi lainnya. Pada umumnya aplikasi-aplikasi yang berada dalam suatu paket mempunyai *interface* yang sama sehingga memudahkan *user* untuk mempelajari dan memakai aplikasi tersebut dan juga aplikasi- aplikasi tersebut mempunyai bisa berinteraksi antar



aplikasi sehingga memudahkan *user*. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolah kata meskipun dibuat pada aplikasi lembar kerja yang terpisah.”

Jadi dapat disimpulkan bahwa aplikasi adalah suatu program dari perangkat lunak komputer yang siap dioperasikan sesuai kebutuhan sehingga memudahkan kegiatan *user*.

2.3.2 Pengertian *E-Commerce*

Wibowo, Edwin Agung dalam Siregar (2016:96) “*Electronic Commerce (e-commerce)* adalah proses pembelian, penjualan atau pertukaran produk, jasa dan informasi melalui jaringan komputer. *ecommerce* merupakan bagian dari *e-business*, di mana cakupan *e-business* lebih luas, tidak hanya sekedar perniagaan tetapi mencakup juga pengkolaborasi mitra bisnis, pelayanan nasabah, lowongan pekerjaan dll. Selain teknologi jaringan *www*, *e-commerce* juga memerlukan teknologi basis data atau pangkalan data (*database*), *e-surat* atau surat elektronik (*e-mail*), dan bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang, dan alat pembayaran untuk *e-commerce* ini”.

Masitah (2018:2) “*E-commerce* atau disebut juga perdagangan elektronik merupakan aktivitas yang berkaitan dengan pembelian, penjualan, pemasaran barang ataupun jasa dengan memanfaatkan sistem elektronik seperti internet ataupun jaringan komputer.”

Dari pernyataan diatas maka dapat disimpulkan bahwa *e-commerce* adalah proses penyebaran, penjualan maupun pembelian barang dan jasa melalui jaringan komputer atau internet.



2.3.3 Pengertian *Website*

Nurmi dalam Hidayat (2017:2) “Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

Abdulloh (2018:1), *Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standard yaitu HTML. Skrip HTML ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang.

Dari kedua definisi diatas dapat penulis simpulkan bahwa *Website* adalah kumpulan halaman yang saling terkait dimana didalamnya memuat konten berupa teks, gambar, animasi atau suara yang dapat diakses pengguna melalui jaringan internet.

2.3.4 Pengertian Aplikasi *E-commerce HD Kitchen* berbasis *Website*

Dari Aplikasi *E-commerce HD Kitchen* berbasis *Website* adalah aplikasi berupa *e-commerce* yang dibuat dengan berbagai fitur yang tujuannya untuk memudahkan penjualan dari HD Kitchen dan dibangun menggunakan pemrograman yang berbasis *website*.



2.4. Teori Program

2.4.1 Pengertian Basis Data (*Database*)

Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi”. Adapun pengertian Basis Data menurut Conolly & Begg dalam Rusmawan (2019:38), “Basis data adalah sebuah kumpulan data yang saling berelasi secara logika dan dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi”. Lain halnya menurut Abdullah (2018:103), “*Database* atau basis data adalah kumpulan informasi yang disimpan dalam computer secara sistematis sehingga dapat diperiksa menggunakan suatu program computer untuk memperoleh informasi”.

Dari beberapa pernyataan diatas, dapat disimpulkan bahwa basis data adalah kumpulan data yang saling berhubungan untuk memenuhi informasi yang tersimpan di penyimpanan komputer secara sistematis agar program komputer dapat terus mengolahnya.

2.4.2 Pengertian *MySQL*

“MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah”. (Saiful & Ambarita, 2017).

Kadir dalam Pranata dkk. (2015:25), “*MySQL* merupakan *software* yang berbasis *structure query language* (SQL) tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *MySQL* adalah sebuah implementasi dari sistem manajemen *database* relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.”



Dapat disimpulkan bahwa *MySQL* merupakan *software* DBMS yang bersifat *Open Source* dan digunakan untuk menyimpan data atau server *database* yang mendukung Bahasa *database* pencarian *SQL*.

2.4.3 Pengertian HTML (*Hypertext Markup Language*)

Madcoms dalam Pranata dkk (2015:26), “HTML (HyperText Markup Language) dikenal sebagai bahasa kode berbasis teks untuk membuat sebuah halaman *web*, keberadaannya dikenal dengan adanya ekstensi *.htm atau *.html.” Adapun menurut Supriyanto dalam Pranata dkk (2015:26), “HTML merupakan suatu bahasa dari *website* (www) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program *browser*.” Lain hal nya Abdulloh (2018:127) meyakini bahwa, “HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*”.

2.4.4 Pengertian PHP (*Hypertext Preprocessor*)

Kadir dalam Pranatak dkk (2015:26), “PHP merupakan singkatan dari “PHP: *Hypertext Preprocessor*” adalah skrip yang dijalankan di *server*. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. Keuntungan PHP, kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi.” Sedangkan menurut Abdulloh (2018:127), *PHP* merupakan kependekan dari *PHP Hypertext preprocessor* yaitu bahasa pemrograman *web* yang dapat disisipkan dalam skrip *HTML* dan bekerja di sisi *server*.” Adapun menurut pendapat Raharjo (2016:38), “PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*”.

2.4.5 Pengertian Web Server

Abdulloh (2018:4), “*Web server* adalah program untuk melayani penyajian halaman *web* di internet. Untuk kebutuhan simulasi, anda dapat menginstal *software web server* di komputer lokal. Jika anda hanya mendesain *web* statis menggunakan *HTML* dan *CSS*, maka anda tidak membutuhkannya. Namun jika



anda mendesain web dinamis menggunakan script pemrograman seperti *PHP*, maka anda membutuhkannya.

Program *web server* populer adalah *Apache*. anda dapat menginstal paket softwarena sesuai system operasi anda. Misalnya untuk windows anda bias memakai WAMP atau XAMPP yang berisi paket *software Apache, MySQL, dan PHP*.”

2.4.6 Pengertian *XAMPP*

Suntoro (2019:11), “XAMPP adalah perangkat lunak yang bersifat *open source*, aplikasi *Apache* (web server) yang mudah di instal dan berisi MariaDB, PHP, dan Perl”. Sedangkan, menurut Santoso dan Nurmalina (2017:86), Xampp merupakan alat bantu yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache, PHP* dan *MySQL* secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis atau auto konfigurasi. XAMPP merupakan paket *PHP* yang berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan XAMPP tidak dibingungkan dengan penginstalan program-program lain, karena semua kebutuhan telah tersedia oleh XAMPP. Yang terdapat pada XAMPP di antaranya : *Apache, MySQL, PHP, FilZilla FTP Server, PhpMyAdmin* dll.

2.4.7 Pengertian *phpMyAdmin*

Standsyah dan Restu (2015:39) “PhpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi database *MySql* melalui jaringan lokal maupun internet.”

2.4.8 Pengertian *Black-Box Testing*

Jaya (2018:45) “Black-Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak.”